In the Claims

No claims are being amended by the present Reply. The following listing of the claims is being provided for the convenience of the Examiner.

1.   (Previously Presented) A method for providing a request to a portlet wherein the portlet renders itself in a graphical user interface (GUI), comprising:

generating a control tree that is a logical representation of the GUI, the control tree including

a portlet control that represents the portlet, and

a set of controls that interact with each other through an event notification mechanism by raising and responding to events,

wherein the set of controls represents graphical and functional elements of the GUI which are related hierarchically to each other;

mapping the request to the control tree;

advancing the control tree through a plurality of stages in a lifecycle based on the request, wherein the lifecycle is defined by a set of stages, and wherein the set of controls in the control tree are allowed to perform certain tasks depending on the control tree's stage in the lifecycle; and

providing the request to a portlet container, wherein the providing is performed by the portlet control.

2.   (Previously Presented) The method of claim 1, wherein:

generating the control tree includes generating the control tree from a factory.

3.   (Previously Presented) The method of claim 1, further comprising:

generating a response wherein the response is used to render at least a portion of the GUI.

4.   (Previously Presented) The method of claim 2 wherein the step of generating the control tree from the factory comprises:

creating a metadata representation of the control tree; and

generating a class to construct the control tree based on the metadata representation.

5.   (Original) The method of claim 1 wherein:

the request is a hypertext transfer protocol request (HTTP); and

the request originates from a web browser.

6.   (Original) The method of claim 3, further comprising:

providing the response to a web browser.

7.   (Previously Presented) The method of claim 1 wherein:

the control tree is driven through the plurality of stages in the lifecycle by an interchangeable lifecycle component.

8.   (Previously Presented) The method of claim 1 wherein:

each one of the set of controls has an interchangeable persistence mechanism.

9.   (Previously Presented) The method of claim 1 wherein:

each one of the set of controls renders itself according to a theme.

10.   (Previously Presented) The method of claim 1 wherein:

each one of the set of controls communicates with another one of the set of controls.

11.   (Previously Presented) The method of claim 1 wherein:

one of the set of controls advances through the plurality of stages in the lifecycle in parallel with another of the at least one controls.

12.   (Previously Presented) The method of claim 1 wherein:

each one of the plurality of stages in the lifecycle is selected from the group consisting of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

13.   (Original) The method of claim 3 wherein:

the response is an hypertext transfer protocol (HTTP) response.

14.   (Canceled).

15.   (Previously Presented) The method of claim 1 wherein:

each one of the set of controls is selected from the group consisting of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

16-28. (Canceled).

29.   (Previously Presented) The method of claim 16 wherein:

each one of the set of controls is selected from the group consisting of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

30.   (Previously Presented) A system for providing a request to a portlet wherein the portlet renders itself in a graphical user interface (GUI), comprising:

a first container operable to map the request to a control tree factory;

the control tree factory operable to generate a control tree based on the request wherein the control tree is a logical representation of the GUI, the control tree including:

at least one portlet control that represents the portlet, and

a set of controls that interact with each other through an event notification mechanism by raising and responding to events,

wherein the set of controls represents graphical and functional elements of the GUI which are related hierarchically to each other;

a lifecycle driver operable to drive the control tree through a plurality of stages in a lifecycle based on the request, wherein the lifecycle is defined by a set of stages, and wherein the set of controls in the control tree are allowed to perform certain tasks depending on the control tree's stage in the lifecycle; and

a portlet container operable to accept the request from the at least one portlet control and provide the request to the portlet.

31.   (Previously Presented) The system of claim 30 wherein:

the portlet generates a response and wherein the response is used to render at least a portion of the GUI.

32.   (Original) The system of claim 30 wherein:

the request is a hypertext transfer protocol request (HTTP); and

the request originates from a web browser.

33.   (Previously Presented) The system of claim 31 wherein:

the response is provided to a web browser.

34.   (Original) The system of claim 30 wherein:

the at least one portlet control has an interchangeable persistence mechanism.

35.   (Previously Presented) The system of claim 30 wherein:

the at least one portlet control renders itself according to a theme.

36.   (Previously Presented) The system of claim 30 wherein:

each one of the at least one portlet controls communicates with another one of the at least one portlet controls.

37.   (Previously Presented) The system of claim 16 wherein:

one of the at least one portlet controls advances through the plurality of stages in the lifecycle in parallel with another of the at least one portlet controls.

38.   (Previously Presented) The system of claim 30 wherein:

each one of the plurality of stages in the lifecycle is selected from the group consisting of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

39.   (Original) The system of claim 31 wherein:

the response is an hypertext transfer protocol (HTTP) response.

- 5 -

40.   (Canceled).

41.   (Previously Presented) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

map a request to a control tree that is a logical representation of a graphical user interface (GUI), the control tree including:

a portlet control that represents a portlet, and

a set of controls that interact with each other through an event notification mechanism by raising and responding to events,

wherein the set of controls represents graphical and functional elements of the GUI which are related hierarchically to each other;

advance the control tree through a plurality of stages in a lifecycle based on the request, wherein the lifecycle is defined by a set of stages, and wherein the set of controls in the control tree are allowed to perform certain tasks depending on the control tree's stage in the lifecycle; and

provide the request to a portlet container, wherein the providing is performed by the portlet control.

42.   (Original) The machine readable medium of claim 41, further comprising instructions that when executed cause the system to:

generate the control tree from a factory based on the request.

43.   (Previously Presented) The machine readable medium of claim 41, further comprising instructions that when executed cause the system to:

generate a response wherein the response is used to render at least a portion of the GUI.

44.   (Original) The machine readable medium of claim 42, further comprising instructions that when executed cause the system to:

create a metadata representation of a control tree; and

generate a class to construct the control tree based on the metadata representation.

- 6 -

45. (Original) The machine readable medium of claim 41 wherein:

the request is a hypertext transfer protocol request (HTTP); and

the request originates from a web browser.

46. (Original) The machine readable medium of claim 43, further comprising instructions that when executed cause the system to:

provide the response to a web browser.

47. (Previously Presented) The machine readable medium of claim 41 wherein:

the control tree is driven through the plurality of stages in the lifecycle by an interchangeable lifecycle component.

48. (Previously Presented) The machine readable medium of claim 41 wherein:

each one of the set of controls has an interchangeable persistence mechanism.

49. (Previously Presented) The machine readable medium of claim 41 wherein:

each one of the set of controls renders itself according to a theme.

50. (Previously Presented) The machine readable medium of claim 41 wherein:

each one of the set of controls communicates with another one of the set of controls.

51. (Previously Presented) The machine readable medium of claim 41 wherein:

one of the set of controls advances through the plurality of stages in the lifecycle in parallel with another of the at least one controls.

52. (Previously Presented) The machine readable medium of claim 41 wherein:

each of the plurality of stages in the lifecycle is selected from the group consisting of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

53. (Original) The machine readable medium of claim 43 wherein:

the response is an hypertext transfer protocol (HTTP) response.

54.   (Canceled).

55.   (Previously Presented) The machine readable medium of claim 41 wherein:

each one of the set of controls is selected from the group consisting of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

56.   (Canceled).

57.   (Previously Presented) A system comprising:

a means for mapping a request to a control tree that is a logical representation of a graphical user interface (GUI), the control tree including:

a portlet control that represents a portlet, and

a set of controls that interact with each other through an event notification mechanism by raising and responding to events,

wherein the set of controls represents graphical and functional elements of the GUI which are related hierarchically to each other;

a means for advancing the control tree through a plurality of stages in a lifecycle based on the request, wherein the lifecycle is defined by a set of stages, and wherein the set of controls in the control tree are allowed to perform certain tasks depending on the control tree's stage in the lifecycle; and

a means for providing the request to a portlet container, wherein the providing is performed by the portlet control.

58.   (Previously Presented) A method for providing a request to a portlet wherein the portlet renders itself in a graphical user interface (GUI), comprising:

generating a control tree that is a logical representation of a GUI of a web application, the control tree including:

a portlet control that represents a portlet , and

a set of controls that interact with each other through an event notification

- 8 -

mechanism by raising and responding to events, wherein the set of controls represents graphical and functional elements of the GUI which are hierarchically related to another and wherein each control in the set of controls is a class in an object oriented programming paradigm that subclasses a superclass providing lifecycle, naming, and child management services;

mapping the request to the control tree;

advancing the control tree through a plurality of lifecycle stages based on the request, wherein the lifecycle is defined by a set of stages, and wherein the set of controls in the control tree are allowed to perform certain tasks depending on the control tree's stage in the lifecycle;

dynamically adding a new control to the control tree;

driving the new control through the plurality of lifecycle stages based on the request until the new control catches up to the control tree's stage in the lifecycle;

providing the request to a portlet container, wherein the providing is performed by the portlet control; and

generating a response based on the request, wherein the response is used to render at least a portion of the GUI.